



UNIVERSITÉ
CAEN
NORMANDIE

Analyse du Jeu de données Aliment en SAS

Auteurs:

Guy Darcy Remesha

Amal Rekik

Bawa Abdoul Mouhsini

Professeur référent:

André Sesboue

3 juillet 2023

REMERCIEMENTS

A travers ces quelques lignes, j'exprime ma profonde gratitude premièrement envers Dieu le Tout-Puissant qui me fait grâce tous les jours et spécialement pour pouvoir réaliser ce travail.

Au terme de ce travail, nous tenons à remercier tout le corps professoral et administratif de l'université de Caen Basse-Normandie. Nos remerciements particuliers s'adressent à notre encadreur Mr André SESBOUE, Enseignant-chercheur et Maitre de conférences, d'abord pour son aide précieux dans la recherche du projet, pour ses conseils, son orientation, sa disponibilité et sa particulière attention qu'il nous a accordée pour que ce travail s'achève. Nous avons beaucoup appris auprès de lui.

Une vive reconnaissance va à l'endroit de Mr Christophe Chesneau, Enseignant-chercheur, maitre de conférences et Responsable du Master 1 SAAD, pour son aide en nous donnant librement les ressources nécessaires à l'élaboration de ce projet.

Enfin, de chaleureux remerciements vont à nos collègues étudiants de la 1ère promotion de l'Institut des Statistiques appliquées qui nous ont soutenus et à toute personne qui a contribué moralement et/ou matériellement pour la réalisation de ce rapport de stage.

Table des matières

1	Cadre générale	Erreur ! Signet non défini.
1.1	Contexte et justification	3
1.2	Objectif de recherche.....	4
1.3	Intérêt du sujet.....	5
1.4	Sources de données	5
1.5	Logiciels utilisés.....	5
2	Indice de silhouette.....	6
2.1	Calcul de l'indice de silhouette	6
2.1.1	Définition.....	6
2.1.2	Déroulement du calcul sur SAS	7
2.2	Représentation graphique de l'indice de silhouette	11
3	Caractérisation des grappes : Parangon.....	12
3.1	Définition.....	12
3.2	Déroulement du programme SAS	12
4	Conclusion.....	14
5	REFERENCES BIBLIOGRAPHIQUES.....	15
6	Annexes	16

1 CADRE GENERALE

1.1 Contexte et justification

Malgré plusieurs méthodes de classification ou de regroupement statistique qui sont disponible, l'un des plus grands défis pour les analystes des données ou la question existentielle qui se pose c'est d'être en mesure d'identifier quelles observations sont éventuellement mal classées parce qu'elles se présentent à la frontière entre deux groupes différents ou en d'autres mots comment comparer deux résultats de regroupement différents.

L'article de 2019 de Ralph Abbey « Comment évaluer différents résultats de clustering » est un bon moyen d'en savoir plus sur les méthodes de clustering et certaines statistiques associées dans SAS.[1]

Parmi plusieurs mesures évoquées dans l'article permettant de comparer les regroupements, il y a la statistique de silhouette considérée comme la plus utile du fait qu'elle permet de mesurer le degré de similitude des données avec la grappe attribuée par rapport aux autres clusters.

En effet, c'est une mesure qui n'est utile que quand il s'agit de la distance euclidienne et que l'on recherche des groupes compacts et clairement séparés [2] et c'est à partir de ces grappes bien formés qu'on arrive à trouver l'élément qui représente mieux les caractéristiques du groupe et qui porte le nom de « Parangon ».

Nombreux logiciels d'analyse statistique (comme R, python, ...) ont déjà inclus la mesure de la silhouette et celle du parangon dans leur environnements de travail où il vous suffit d'appeler la fonction portant le nom tout en respectant les mesures et conditions d'utilisation de ces derniers.

Système d'analyse statistique "SAS" qui est un langage de quatrième génération publié par l'Institut Sas depuis 1976. Depuis 2004, SAS utilise la version 9.[3]

C'est la même version que nous allons utiliser pour calculer les différentes statistiques, à base des procédures existantes déjà et quelques programmes, qui n'a jusqu'à présent inclus ces deux statistiques dans son environnement de travail.

Comme problématique, ce projet traite l'indice de silhouette et le parangon:

Comment calculer ces deux statistiques sur SAS?

Comment les tracer sur SAS?

Qu'est ce qu'on va pouvoir interpréter à partir des résultats trouvés?

Pour illustration, nous allons utiliser un jeu de données « aliments » et nous allons comparer les résultats avec celles de la polycopié de M.Christophe Chesneau.[4]

1.2 Objectif de recherche

L'objectif général de cette recherche est de faire une classification statistique des nutriments présents dans différentes denrées alimentaires et plus spécifiquement :

- Déterminer les denrées alimentaires par groupe en tenant compte de la quantité de l'énergie contenu, de la quantité des protéines, de la quantité de matières grasses, de la quantité du calcium et du fer contenu par le nutriment.
- Déterminer l'indice de silhouette pour chaque nutriment et confirmer la robustesse des regroupements faits.
- Déterminer l'aliment le mieux représenté dans chaque groupe.

1.3 Intérêt du sujet

Ce travail présente un double intérêt :

- ❖ Au niveau académique, il sera utile pour toute personne amenée à faire la classification statistique plus précisément sur le calcul du parangon, de l'indice de silhouette ainsi que sa représentation. Il servira à enrichir la documentation pour d'autres chercheurs à venir et ceux de l'Université de Caen Normandie du master SAAD en particulier.
- ❖ Au niveau personnel, il nous a permis d'appliquer les connaissances théoriques apprises au cours du cursus académique, d'en acquérir de nouvelles dans le domaine de la nutrition et de nous familiariser au monde de la recherche.

1.4 Sources de données

Pour tester les hypothèses d'étude, nous avons utilisé le jeu de données aliments se trouvant dans le livre « CHAPMAN & HALL/CRC A CRC Press Company A Handbook of Statistical Analyses using SPSS » publié en 2004. [5]

En résumé, le jeu de données comprend des informations sur les aliments, y compris leur nom, leurs caractéristiques nutritionnelles telles que les calories, les protéines, les graisses, le calcium et le fer. Ces variables fournissent des informations essentielles pour l'analyse de classification et la catégorisation des aliments en fonction de leurs valeurs nutritionnelles.

1.5 Logiciels utilisés

Le traitement, l'analyse des données et la représentation graphique ont été réalisés grâce au logiciel statistique et langage de programmation « statistical analysis software » SAS en sigle. Les résultats ont été présentés sous forme de tableaux et de graphiques.

2 Indice de silhouette

La statistique de la silhouette a été introduite par Peter Rousseeuw et est conceptuellement facile à comprendre car il est basé sur la géométrie des clusters.[6]

La statistique de silhouette est calculée pour chaque observation et mesure dans quelle mesure l'observation s'intègre dans son cluster attribué.

Qualitativement, une statistique de silhouette proche de 1 signifie que l'observation est située au centre de son groupe attribué. Une statistique négative signifie que l'observation est située à l'intérieur d'un autre groupe et qu'elle est donc potentiellement mal attribuée. De cette manière, la mesure de silhouette vous permet d'évaluer les observations individuelles et la qualité globale d'une affectation de clustering. Vous pouvez calculer la mesure de silhouette pour l'ensemble de données complet en faisant la moyenne des valeurs de silhouette sur toutes les observations. De cette manière, la mesure de silhouette vous permet d'évaluer les observations individuelles et la qualité.

2.1 Calcul de l'indice de silhouette

2.1.1 Définition

Pour tout $i \in \{1, \dots, n\}$, on appelle indice de silhouette associé à l'individu w_i le réel :

$$s(i) = \frac{b(i) - a(i)}{\max \{a(i), b(i)\}}$$

$a(i)$ est la moyenne des distances entre w_i et les individus de son groupe.

$b(i)$ est la moyenne des distances entre w_i et les individus du groupe le plus proche de celui auquel il appartient.

Il faut noter que lorsqu'un groupe ne contient qu'un seul objet, la définition de n n'est pas claire. Nous fixons simplement $s(i)$ à zéro. Ce choix est bien sûr arbitraire, mais une valeur de zéro semble être la plus neutre.[5]

2.1.2 Déroulement du calcul sur SAS

Les statistiques de silhouette sont des mesures basées sur la distance. Il compare la distance moyenne entre un point et d'autres points dans le même cluster avec la distance moyenne entre ce point et des points dans d'autres clusters. Je montrerai comment calculer des statistiques à l'aide du langage de programmation SAS. Il est logique d'appliquer d'abord l'algorithme de classification hiérarchique :

1. Premièrement, grâce à l'appel de la procédure proc distance en mentionnant l'option incluse, « method= euclid », nous calculons la matrice des distances euclidiennes et on nomme le nom du tableau de sortie.
2. Après, nous appliquerons le regroupement sur le jeu de données initial grâce à la procédure cluster. Nous avons utilisé la méthode des distance moyenne « method=average ».
3. Ensuite nous prenons la sortie de la procédure cluster et l'appliquons à la procédure Tree pour indiquer sas le nombre de regroupement que l'on souhaite.
4. Avec la sortie de la procédure Tree, nous garderons les aliments et leur clusters respectives dans une étape data.

Si i est un point dans l'ensemble de données, vous calculez la valeur de la silhouette en i en effectuant les calculs suivants :

- Avant le début de l'analyse, il faut ordonner par ordre alphabétique tous les aliments
- Créer un jeu donné contenant les aliments en variable mais étiqueté autrement avec leur cluster du fait que l'appel de 2 tableaux ayant les mêmes variables lors d'une étape data est très compliqué sur SAS.

- Dans une étape data, nous appelons le jeu de données contenant les aliments (étiquetés autrement) en variable et les clusters en observation et nous appelons également le jeu de données ayant comme variable les aliments et comme observation les distances euclidiennes de chaque aliment avec les autres aliments (c'est une forme de matrice carrée).
- Définissez une variable $a(i)$ comme étant la distance moyenne entre p et les autres points du même cluster. Il s'agit de la distance moyenne intra-cluster. Le calcul de ce paramètre est dans la condition, se trouvant dans la boucle allant de 1 à la dimension de la table des distances, qui compare l'élément le cluster de l'élément $[_n_]$ et le cluster de la i ème observation de la table des distances.
- Définissez un tableau b dont la taille est le nombre de cluster (3). Si pour chacun des autres clusters, elles sont trouvées si la condition précédente n'est pas vérifiée et nous pouvons trouver la distance moyenne entre i et les points de l'autre cluster. S'il y a k clusters, cela produit $k-1$ valeurs. Définir $b_min(i)$ comme étant le MINIMUM des valeurs du tableau b contenant les distances moyennes inter-clusters. C'est le minimum de la distance moyenne entre les clusters.

Il faut noter que le tableau b est initialisé sur valeur manquante et il change si la condition de comparaisons des clusters n'est pas vérifié.

S'il arrive que le paramètre « a » n'est pas calculé pour une observation donnée cela veut dire l'observation est toute seule dans son regroupement et il définit directement la statistique de silhouette à zéro.

Dans le cas contraire, il définit la statistique de silhouette comme

$s(i) = (b_min(i) - a(i)) / \max(b_min(i), a(i))$. C'est une valeur dans l'intervalle $[-1, 1]$.

L'illustration de ce code sur un jeu de données nous donne les résultats suivants :

Obs.	aliment	cluster_indiv	a	b1	b2	b3	b_min	s
1	BB	1	43.011	.	205.612	392.593	205.612	0.79081
2	BR	1	100.971	.	284.650	433.762	284.650	0.64528
3	BS	1	61.043	.	239.862	408.322	239.862	0.74551
4	HR	1	97.640	.	114.512	363.942	114.512	0.14734
5	HS	1	43.011	.	205.612	392.593	205.612	0.79082
6	LL	1	79.941	.	133.420	368.122	133.420	0.40083
7	LS	1	57.881	.	167.009	377.937	167.009	0.65342
8	PR	1	43.159	.	205.721	392.649	205.721	0.79020
9	PS	1	48.477	.	220.291	399.048	220.291	0.77994
10	AC	2	121.414	295.160	.	323.066	295.160	0.58865
11	AR	2	104.558	273.009	.	305.814	273.009	0.61701
12	BC	2	66.103	152.420	.	350.003	152.420	0.56631
13	BH	2	63.331	172.867	.	353.628	172.867	0.63364
14	BT	2	83.551	126.884	.	360.924	126.884	0.34152
15	CB	2	76.136	217.515	.	364.893	217.515	0.64997
16	CC	2	64.046	162.575	.	355.161	162.575	0.60606
17	FB	2	64.400	198.271	.	344.989	198.271	0.67519
18	HF	2	66.047	197.556	.	354.944	197.556	0.66568
19	MB	2	80.244	132.014	.	362.590	132.014	0.39215
20	MC	2	133.088	233.458	.	211.569	211.569	0.37095
21	PF	2	74.591	137.303	.	353.377	137.303	0.45675
22	RC	2	93.155	240.773	.	278.076	240.773	0.61310
23	SC	2	134.709	261.891	.	216.583	216.583	0.37802
24	TC	2	84.531	244.369	.	341.258	244.369	0.65408
25	UC	2	65.777	162.556	.	360.159	162.556	0.59536
26	VC	2	69.766	147.347	.	358.036	147.347	0.52652
27	DC	3	.	392.107	329.122	.	.	0.00000

Le tableau ci-dessus montre la colonne aliment qui contient tous les aliments, la colonne cluster_indiv qui contient les clusters de chaque aliment, la colonne a qui contient la distance moyenne de l'individu avec les autres éléments du même groupe, les colonnes b1 b2 et b3 qui renferment les distances moyennes de chaque individu avec les autres éléments du cluster différent de l'individu en question (distance moyenne inter-cluster).

Il faut noter que les colonnes b1, b2 et b3 prennent la valeur vide que lorsqu'il rencontre un élément de son propre groupe ce qui revient à calculer bien évidemment la distance moyenne entre les individus et les autres éléments du même cluster.

La colonne b_min prend la distance moyenne minimale entre les colonnes b1, b2 et b3. Enfin la dernière colonne « s » contient l'indice de silhouette pour chaque aliment calculé à l'aide des paramètres a et b_min.

Pour rappel également, si dans un cluster il y a une et seule observation, l'indice de silhouette est nul pour l'individu ce qui est le cas pour l'aliment DC.

Globalement, du fait qu'aucun indice de silhouette n'est négatif, cela nous amène à conclure que chaque individu est dans le bon groupe et qu'aucun élément ne pourra être déplacé dans le groupe le plus proche.

2.2 Représentation graphique de l'indice de silhouette

Grâce à la procédure sgplot et à l'option « hbarparm », un graphique en barres est fait pour afficher la distribution des silhouettes pour chaque cluster.

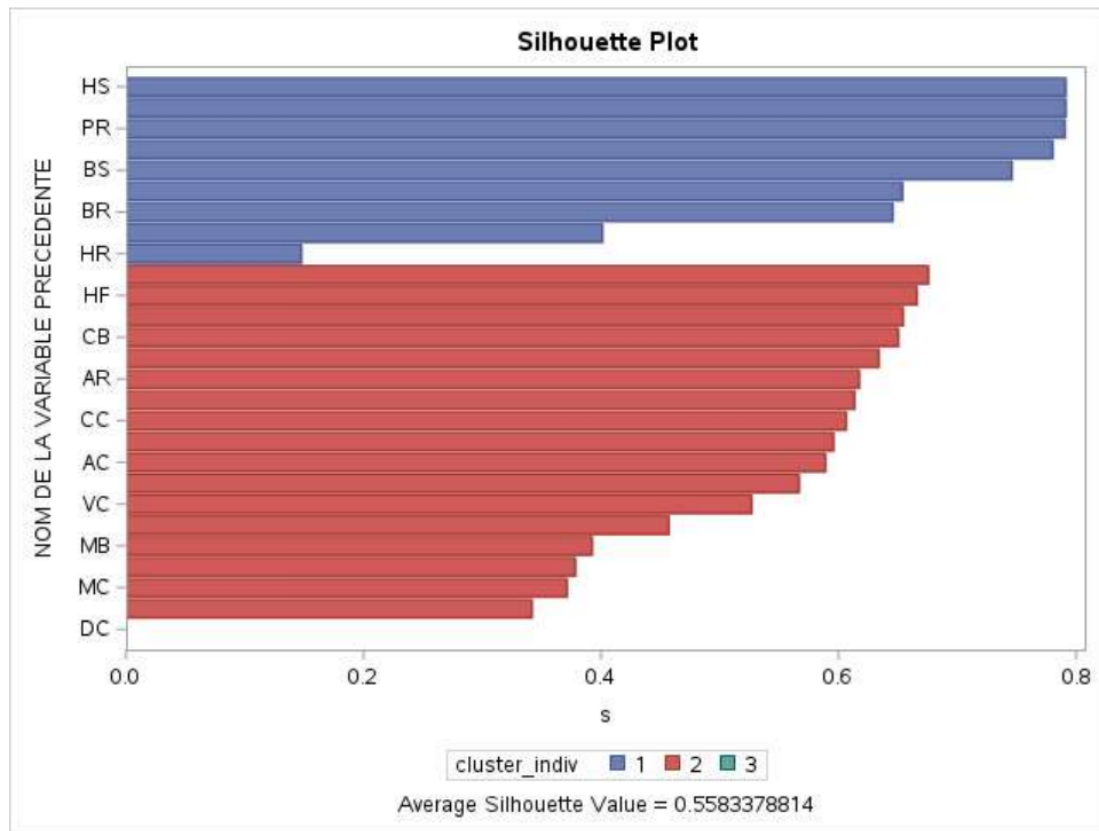


Figure 4 : représentation graphique de l'indice de Silhouette

Le graphique à barres contient les informations du calcul de la valeur silhouette que le tableau précédent, sauf que les observations au sein de chaque cluster sont triées par ordre décroissant de la valeur de la silhouette.

Nous constatons qu'en moyenne la valeur de l'indice de silhouette est à peu près 0.56, ce qui est confirmé par les résultats de l'analyse du même jeu de données sous R dans la polycopie « Classification » de Christophe Chesneau [7], et prouve que c'est une partition assez forte pour garder le regroupement.

3 Caractérisation des grappes : Parangon

3.1 Définition

Statistiquement parlant, l'individu le plus proche du centre de gravité du cluster porte le nom de parangon.

Le profil de cet individu caractérise alors le groupe auquel il appartient.

3.2 Déroulement du programme SAS

Jusque-là, il n'y a pas de procédure ou fonction prédéfini pour calculer le parangon, qui est l'observation le plus proche du centre du cluster sur SAS ce qui nous ramène aussi à passer par une étape data en utilisant des boucles pour parcourir tous les individus en tenant compte de leur grappe(cluster).

En effet, il faut tout d'abord :

- ❖ Calculer les moyennes par regroupement (cluster) et garder les résultats dans un tableau grâce à la procédure means.
- ❖ Fusionner le jeu de données provenant de la procédure tree et le jeu de données ayant de chaque variable par cluster.
- ❖ Recalculer la matrice des distances sur le nouveau jeu de données formes.
- ❖ Sélection de tous les centres de gravité sur la colonne des variables, de tous les aliments comme observation et transposer le résultat.

Si i est un point dans le nouveau jeu de données (Centre), vous calculez la valeur du parangon en i en effectuant les calculs suivants :

- ❖ Dans une étape data, nous créons un jeu donné contenant les aliments comme des variables et les centres de cluster comme observation.
- ❖ Nous définissons une variable minimum qui recherche le minimum sur chaque ligne d'observation.

- ❖ A partir d'une boucle ayant comme dimension le jeu de données, nous initialisons une condition pour comparer si le ième élément est égale au minimum et enfin nous définissons une variable `paragon_par_cl` qui garde la variable ayant l'élément minimum grâce à la fonction `vname`.

Obs.	aliment	minimum	parag_par_clas
1	c1	8.9589	HS
2	c2	20.8954	FB
3	c3	0.0000	DC

L'aliment HS « Smoked ham » est celui qui représente le premier groupe car il est plus proche du centre du cluster 1 (c1).

L'aliment FB « Bluefish, bake » est celui qui représente le second groupe formé car il est plus proche du centre du cluster du cluster 2 (c2).

Enfin l'aliment DC « Sardines, canned » est celui qui représente le dernier groupe formé car il est plus proche du centre du cluster du cluster 3 (c3).

4 **Conclusion**

Dans le cadre de notre analyse de classification des aliments, nous avons utilisé la méthode de clustering pour regrouper les aliments en différents clusters.

Après avoir calculé les centres de classe, nous avons identifié les aliments les plus proches de ces centres, qui représentent les exemples les plus représentatifs de chaque classe.

Les centres de classe ont fourni une vision globale des caractéristiques moyennes de chaque groupe d'aliments, ce qui nous a permis de mieux comprendre les différences entre les clusters.

En utilisant la distance euclidienne, nous avons identifié les parangons, c'est-à-dire les aliments les plus similaires aux caractéristiques moyennes de chaque classe.

Ces parangons sont des exemples représentatifs de leur classe respective, fournissant ainsi des informations précieuses sur les caractéristiques clés qui définissent chaque groupe d'aliments.

Notre analyse de classification des aliments a permis d'identifier des clusters distincts et de mettre en évidence les aliments les plus représentatifs de chaque classe.

Ces résultats fournissent des informations précieuses pour comprendre les différences entre les groupes d'aliments et peuvent être utilisés dans divers contextes liés à la nutrition, la santé et la commercialisation des produits alimentaires.

5 REFERENCES BIBLIOGRAPHIQUES

[1] Ralph Abbey <https://support.sas.com/resources/papers/proceedings19/3409-2019.pdf>.

[2] Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Peter J. ROUSSEUW.

[3] Wikipedia [https://fr.wikipedia.org/wiki/SAS_\(langage\)](https://fr.wikipedia.org/wiki/SAS_(langage))

[4] Eléments de classification d Christophe CHESNEAU, <https://chesneau.users.lmno.cnrs.fr/classif-cours.pdf>

[5] CHAPMAN & HALL/CRC A CRC Press Company Un manuel d'analyses statistiques utilisant SPSS <https://www.academia.edu/34413004>

6 Annexes

```
/****** importation des donnees les donnees
******/

data aliments;
  infile "&home/aliments.csv" firstobs=2 dlm=';';
  input aliment$ aliment_n1 :$16. X1 X2 X3 X4 X5; *X1-X5;
  label X1="Nombre de calories"
        X2="Teneur en protéines"
        X3="Teneur en graisse"
        X4="Teneur en calcium"
        X5="Teneur en fer";
run;

proc print data=aliments label noobs; run;
/******=====
=====***/

/*Trier les donnees par ordre alphabetique de la variable aliment*/
proc sort data=aliments; by aliment; run;

/*Calcul des distances entre observation*/
proc distance data=aliments method=euclid out=fulldistance shape=sqr;
  var interval(x1--x5);
  id aliment;
run;
proc print data=fulldistance;run;
/******=====
=====***/

/****** Application de l'algorithme de CHA
******/
proc cluster data=aliments outtree=dist_aliments method=average
nonorm;
  id aliment;
run;
proc print data=dist_aliments;run;
/******=====
=====***/

/*Regrouper les donnees de la classification*/
%let nclus = 3; /* le nombre de cluster */
```

```

proc tree data=dist_aliments out=dist_clus nclusters= &nclus;
    id aliment;
    copy X1--X5;
run;

proc sort data=dist_clus;by aliment;run;
proc print data=dist_clus; run;

data clusters;
    set dist_clus (keep = aliment cluster);run;

/***** Afficher les clusters sous forme de variables
*****/
proc transpose data=clusters out=clusters(rename=(_NAME_=aliment))
prefix=i;
run;
proc print data=clusters;run;

/***** Calcul de la valeur du silhouette pour chaque observation
*****/
data silhouette;
    if _n_ = 1 then do;
        set clusters;
        array cluster{*} i1-i27;
    end;
    set fulldistance;

    array distance{*} AC--VC;

    cluster_indiv = cluster[_n_]; /*prends le cluster de chaque
observation*/
    a = 0; /*la moyenne des distances entre wi et les individus de
son groupe*/
    cardinal_a = 0; /*Variable qui va compter tous les elements du
meme cluster */

    array b[&nclus];
    array cardinal_b[&nclus]; /*tableau de variable qui va compter
tous les elements du meme cluster */

    do k = 1 to dim(b);
        b[k] = 0;

```

```

        cardinal_b[k] = 0;
    end;

    b{cluster_indiv} = .;
    cardinal_b{cluster_indiv} = 1;

    do i = 1 to dim(distance);
        if distance[i] ge 0 then do;
            if cluster[_n_] = cluster[i] then do;
                a = a + distance[i];
                cardinal_a = cardinal_a + 1;
            end;
        else do;
            b[cluster[i]] = b[cluster[i]] +
distance[i];
            cardinal_b[cluster[i]] =
cardinal_b[cluster[i]] + 1;
        end;
    end;

    end;

    a = a / (cardinal_a - 1);
    do j = 1 to dim(b);
        b[j] = b[j] / cardinal_b[j];
    end;

    if a ^= . then do;
        b_min = min(of b{*});
        s = (b_min - a) / max(of a,b_min);
    end;
    else do;
        s = 0;
    end;

    keep aliment s cluster_indiv a b1 b2 b3 b_min;

run;
/*===== Fin du calcul de la silhouette=====*/

/***** Tri de l'indice des observations par cluster et par la valeur
du silhouette*/
proc sort data=silouhette out=silouhette;
by cluster_indiv descending s;

```

```

run;
proc print data =silouhette;run;
/*****=====
=====*****/

/* calcul de la moyenne de la silhouette et stockage de la valeur
dans une macro variable appele s_mean */
proc means data=silouhette mean;
  var s;
  output out = s_mean mean(s) = smean;
run;

data s_mean;
  set s_mean;
  call symputx('s_mean', smean);
run;
/*=====
=====*/

/*****Representation graphique de l'indice de
silhouette*****/
title "Silhouette Plot";
footnote "Average Silhouette Value = &s_mean";
proc sgplot data=silouhette;
  hbarparm category=aliment response=s/ group=cluster_indiv;
run;
/*-----
-----*/

/*****Recherche du
parangon*****/
proc means data=dist_clus maxdec=3 mean noprint; class cluster;
  output out= centre (DROP=_TYPE_ _FREQ_) mean(X1-X5) = X1-X5 ;
run;
proc print data=centre ;run;

data centre;
  set centre;
  IF cluster = "." THEN DELETE;
run;
proc print data=centre ;run;

```

```

data centre;
    set centre;
    LENGTH aliment $3.;
    if cluster = 1 then aliment = "c1";
    if cluster = 2 then aliment = "c2";
    if cluster = 3 then aliment = "c3";
    drop cluster;
run;

/* Réordonner les variables et Affichage du tableau de données */
DATA centre;
RETAIN aliment X1 X2 X3 X4 X5;
SET centre;
RUN;
PROC PRINT DATA = centre;
RUN;

/*Fusion des deux datasets*/
data aliments_cent;
    set aliments centre;
run;

/* calcul de la distance euclidienne entre le nouvelles individus*/
proc distance data=aliments_cent method=euclid out=fulldist_nouv
shape=sqr;
    var interval(x1--x5);
    id aliment;
run;
proc print data=fulldist_nouv; run;
/*-----*/
-----*/

/*Recherche de l'element et la distance le plus proche de chaque
centre de classe*/
/* Selection des 27 premiers observations et des 3 derniers
colonnes(c1, c2 et c3)****/
data parangon;
    set fulldist_nouv (firstobs=1 obs=27 keep= aliment c1--c3) ;
    *array dist_cent{*} _numeric_;
run;

```

```

/*Transpose de la table parangon pour chercher le minimum sur chaque
ligne*/
proc transpose data=parangon out=transposed_data
(rename=(_NAME_=aliment ));
  var c1 c2 c3;
  id aliment;
run;

/* Trouver l'aliment correspondant à la valeur minimale et le nom de
cet aliment */
data minimum_data;
  set transposed_data;
  array dist_cent_new{*} AC--VC;

  minimum = min(of AC--VC);

  do i =1 to dim(dist_cent_new);
    if dist_cent_new[i] = minimum then do;
      parag_par_clas = vname(dist_cent_new(i));
      leave;
    end;
  end;

  keep aliment minimum parag_par_clas;
run;
/*une boucle do pour parcourir les variables numériques et vérifier si
elles correspondent à la valeur minimale minimum.
Lorsque nous cherchons une correspondance, nous utilisons la fonction
vname pour obtenir le nom de la variable correspondante,
puis nous l'assignons à la variable parag_par_clas .*/

/* Afficher le résultat */
proc print data=minimum_data;
run;
/*-----*/
-----*/

```